

Drawing Tablet Support in Wayland



Who am I?

- My name is Stephen Chandler Paul
- I go by the alias 'Lyude'
- This is my first major contribution to the Wayland project
- Currently a Sophomore in college.

The Current State of Tablet Support

- What's working:
 - Basic motion support
 - Tool objects
 - Pressure
 - Distance
 - Tilt
 - Stylus buttons

The Current State of Tablet Support

(cont.)

- What isn't working:
 - Active area selection
 - Rotation
 - Tablet mice
 - Pad buttons, selection wheels, touch strips, etc.
 - LEDs

What's so challenging about drawing tablets?

- Drawing tablets are **weird**
- They have far more information and buttons than most input devices do
 - Tons of different tools, mice, airbrushes, pens, pencils, etc.
 - They give far more information than any other type of input tool (tons of axes, tons of buttons, tool serial numbers, etc.)
 - Multiple tablets can be used with the same system, so they have to all be handled separately

libinput and tablets

libinput before the GSoC

- No real tablet support
- Patches had been submitted for it (courtesy of Carlos Garnacho)
 - Original idea was to have tablets just be special mice
 - This was the same approach Windows and OS X use, and the same approach Xorg currently uses.

Tablets in libinput now

- Tablets are handled separately from mice
- This means that clients have to handle them separately too, a disadvantage
- But the pros outweigh the cons:
 - We avoid complicating the API for mice
 - Easier to maintain
 - While pushing everything through the cursor is a smart approach for legacy applications, Wayland is modern enough that this shouldn't be necessary.

Tablet Axes in libinput

- Every time libinput receives an EV_SYN from evdev, if any of the axes values have changed we send the caller a tablet event event with the current state of each axis, along with a bitfield indicating which axes changed
- We still include the last known values of axes that haven't updated
- [Most] axes are normalized
 - This might change in the future if we can find reliable units to return the axis values in
- The axes for tool movements are included in these events

Normalization of Tablet Axes

- Currently, all tablet axes (except for X and Y) are normalized
- There isn't any real unit we can translate pressure into
- The X and Y axes for tilt can supposedly be translated into degrees, this just hasn't been done yet
- Theoretically, distance could potentially be translated into mm. The problem is we don't have any real guarantees on the accuracy of the sensor, which seems to change depending on the location of the tablet tool above the tablet's surface.
- Ideally we would like to avoid using normalization on all of these axes, but right now for the most part we don't really know how to

The `libinput_tool` object

- Represents a physical tool that is, or has been in use with a drawing tablet, such as
 - Pens
 - Pencils
 - Airbrushes
 - Mice
 - Mice with lenses
- Contains the physical type of tool and, if applicable, it's serial number
- User data can be associated with each object, using the `libinput_tool_get_user_data()` and the `libinput_tool_set_user_data()` functions

The `libinput_tool` object

(continued)

- On tablets that report serial numbers for physical tools, `libinput_tool` objects can be used to differentiate multiple physical tools, even if they're of the same type
- The lifespan of a `libinput_tool` extends beyond the duration it's tool is in proximity.
- So, whenever the tool comes back into proximity, it's always guaranteed to have the exact same `libinput_tool`
- Every tool that is used with a tablet is stored in a global tool list in the `libinput` seat structure

Tablets that don't report serial numbers

- Certain tablets, namely the current Intuos and Bamboo tablets, don't report serial numbers.
- Without serial numbers, we cannot make any guarantee that each `libinput_tool` object is associated with a unique tool
- Tools that come into proximity of tablets like these are stored in a list of tools associated with the tablet as opposed to the seat
- They stay valid for as long as the tablet is connected, and are discarded when the tablet disconnects
- If desired, a program can extend the lifetime by calling `libinput_tool_ref()`

Buttons

- Right now, only stylus buttons are reported
- linuxwacom is very inconsistent with tablet buttons
- Some devices report presses on the tablet through the pad device, some of them report presses through the stylus device
- Work is being done on improving linuxwacom, so it's not worth hassling with these complications yet until the driver's been sorted out



Touch wheels/touch strips

- Consistent (I think?), but also messy
- Touch wheels (the one on the Intuos pro specifically) share axes that are used by the stylus for other things
- Indicates data is coming from the touch wheel/strip by reporting the serial number (MSC_SERIAL) of the tool as being -1
- Again, this behavior is being fixed up in linuxwacom

Button boxes

- One of the proposed solutions for buttons on drawing tablets
- Represents any device that's dedicated to housing a lot of buttons, selection wheels, etc.
- Useful because the shortcut buttons on Wacom can vary greatly across devices
- Good for other devices such as the Novation Launchpad, Griffin PowerMate, etc.

The Wayland Protocol for Tablet Devices

Status of Tablet Support in the Wayland Protocol

- Has not been merged upstream yet
- We have a working Weston implementation, along with a small demo application for tablets
- All of the features we support in libinput are currently in the protocol, and are functional in the current Weston implementation
- Basic support for using tablets with the Weston desktop-shell is supported with the exception of resizing windows
- Most other Weston applications still need support added for tablets

The three new objects

- `wl_tablet_manager`
- `wl_tablet_tool`
- `wl_tablet`

wl_tablet_manager

- Tablets cannot be multiplexed into a single input device
- Global object, similar to the wl_seat object
- wl_tablet_manager is responsible for notifying clients of new tablets and tablets being removed
- Informs clients about each tablet, e.g. the manufacturer, model, etc.

wl_tablet

- Responsible for actually sending all of the input events

wl_tablet_tool

- Represents a physical tool in use with the tablet, just like `libinput_tool`
- Contains the physical type of tool, along with it's capabilities
 - Certain tools have different capabilities in terms of which axes are reported
- Sends a remove event when the lifespan of the tool is expected to end (e.g. the tablet for a serial-less tool disconnects from the system)

Tablet Axes in the Wayland protocol

- Grouped into various different events:
 - `wl_tablet::motion`
 - `wl_tablet::pressure`
 - `wl_tablet::distance`
 - `wl_tablet::tilt`
- There is also `wl_tablet::frame`
 - Used to mark the end of a series of axis updates, analogous to how axis events are grouped in libinput and the `EV_SYN` event in evdev
- Tablets are normalized from either 0 – 65535 or -65535 – 65535, since keeping the numbers as-is when converting to fixed-width would result in too significant of a loss of accuracy
- Using 65535 also gives us a lot of leeway for avoiding overflow

Proximity in/out

- Proximity in and out events are both indicated with
 - `wl_tablet::proximity_in`
 - `wl_tablet::proximity_out`
- These events are also used for indicating that the tablet tool has simply left the proximity of the surface it was hovering over previously
- Contain the ID of the current `wl_tablet_tool` in use

wl_tablet::set_cursor

- Sets the current cursor being used to indicate the tablet tool's location, analogous to setting the cursor for the mouse cursor in Wayland
- Even with tablets with built in screens, having some sort of cursor to correspond with the tablet tool is very important
- Many tablets have issues with accuracy, so the user always needs to know where the computer thinks the pen actually is
- In Weston, the cursor disappears once the tablet tool leaves proximity

wl_tablet::button

- Indicates when a button has been pressed or released
- Buttons stay pressed down until they're actually released, even if the tool travels to a different surface (same behavior as mouse buttons)
- If the tool leaves physical proximity of the tablet, the buttons are marked as released

wl_tablet::down/up

- Used to indicate when a tablet tool begins or finishes physical contact with the tablet
- Same behavior as wl_tablet::button, the tool is not marked as up until it goes up or leaves proximity of the tablet

How wl_tablet_tools are handled

- When a tool that a client hasn't seen before comes into use with a tablet that the client is listening to events from, a `wl_tablet_manager::tool_added` event is sent.
- Clients aren't notified of a tool unless it comes into proximity of a surface they own, so notifications for preexisting tool objects aren't sent right away
- Lifetime of a `wl_tablet_tool` is the same as the lifetime of a `libinput_tool`; forever for normal tablets, destroyed on disconnection for tablets that don't report serial numbers

How multiple tablets are handled

- When a tablet is connected or disconnected, `wl_tablet_manager` notifies each client that has setup a listener with it
- When a client connects, it is sent a `wl_tablet_manager::device_added` event for each tablet that is currently connected to the system

Binding Tablets to Screens

- Right now, all tablets (Cintiqs and other tablets with built in displays included) are binded to the primary screen
- We need to come up with heuristics to determine the characteristics of each tablet:
 - The type of tablet:
 - Tablet without screen
 - Tablet with built-in screen
 - Tablet with built-in screen on device
 - What screen it should be associated with

Active Area Selection

- Allows the user to change the area of the tablet where the tool is considered in proximity
- Useful for cropping the edges off tablets when trying to scale coordinates for a tablet whose ratio is significantly different from the display ratio of the monitor
- We need the user to be able to control this
 - Some users may not care about ratio scaling to a certain degree, e.g. they may not care about the small accuracy lost when scaling to 16:9 on a 16:10 tablet
 - Manual control of the active area on the tablet is already a feature on other operating systems, notably Windows and Macintosh